QUANTIZATION LOOP WITH HEURISTIC APPROACH

TECHNICAL FIELD

The present invention relates to a quantization loop with a heuristic approach. The heuristic approach reduces the number of iterations necessary to find an acceptable quantization threshold in the quantization loop.

BACKGROUND OF THE INVENTION

A computer processes audio or video information as numbers representing that information. The larger the range of the possible values for the numbers, the higher the quality of the information. Compared to a small range, a large range of values more precisely tracks the original audio or video signal and introduces less distortion from the original. On the other hand, the larger the range of values, the higher the bit-rate for the information. Table 1 shows ranges of values for audio and video information of different quality levels, and corresponding bit-rates.

Information type and quality	Range of values	Bits
Video image, black and white	0 to 1 per pixel	1
Video image, gray scale	0 to 255 per pixel	8
Video image, "true" color	0 to 16,777,215 per pixel	24
Audio sequence, voice quality	0 to 255 per sample	8
Audio sequence, CD quality	0 to 65,535 per sample	16

Table 1: Ranges of values and bits per value for different quality audio and video information

High quality audio or video information has high bit-rate requirements. Although consumers desire high quality information, computers and computer networks often cannot deliver it.

To strike a balance between quality and bit-rate, audio and video processing techniques use quantization. Quantization maps many values in an analog or digital signal to one value. In an analog signal, quantization assigns a number to points in the signal. In a digital input signal with a range of 256 values, quantization can assign instead one of 64 values to each point in the signal. (Values from 0 to 3 in the input signal are assigned to the quantized value 0, values from 4 to 7 are assigned to the quantized value 1, etc.) To reconstruct the original value, the quantized value is multiplied by the quantization factor. (The quantized value 0 reconstructs 0 x 4=0, the quantized value 1 reconstructs 1 x 4=4, etc.) In essence, quantization decreases the quality of the signal in order to decrease the bit-rate of the signal. After a value has been quantized, however, the original value cannot always be reconstructed. (If the values from 0 to 3 are assigned to the quantized value 0, for example, on reconstruction it is impossible to determine if the original value was 0, 1, 2, or 3.)

When quantizing an input signal, several factors affect the result. For an analog signal, a dynamic range sets the boundaries of the quantization. Suppose the range of an analog signal stretches from negative infinity to infinity, but almost all information is close to zero. The dynamic range of the quantization focuses the quantization on the range of the signal most likely to yield information. For an input signal already in digital form, the dynamic range is bounded by the lowest and highest possible values.

Within the dynamic range, the number of quantization levels determines the precision with which the quantized signal tracks the original signal, which

affects the distortion of the quantized signal from the original. For example, if a dynamic range has 256 quantization levels, each point in an input signal is assigned the closest of the corresponding 256 values. Increasing the number of quantization levels in the same dynamic range increases precision and decreases distortion from the original, but increases bit-rate. Quantization threshold, or step size, is a related factor that measures the distance between quantized values.

The preceding examples describe uniform, scalar, non-adaptive quantization -- each point in the input signal is quantized by the same quantization threshold to produce a single quantized output value. Other quantization techniques include non-uniform quantization, vector quantization, and adaptive quantization techniques. Non-uniform quantization techniques apply different quantization thresholds to different ranges of values in the input signal, which allows greater emphasis to be given to ranges with more information value. Vector quantization techniques produce a single output value representing multiple points in the input signal. Adaptive quantization techniques change dynamic range, the number of quantization levels, and/or quantization thresholds to adapt to changes in the input signal or resource availability in the computer or computer network. For more information about quantization and the factors affecting the results of quantization, see Gibson et al., Digital Compression for Multimedia, "Chapter 4: Quantization," Morgan Kaufman Publishers, Inc., pp. 113-138 (1990).

Some adaptive quantization techniques vary dynamic range while holding constant the number of quantization levels. These techniques adapt to the input signal to maintain a relatively constant degree of quality, and they produce a

relatively constant bit-rate output. One goal of these techniques is to minimize distortion between the input signal and quantized output for the number of quantization levels. Another goal is to optimize entropy, or information value, of the quantized output. The entropy of the quantized output predicts how effectively the quantized output will later be compressed in entropy compression.

Entropy is a useful measure, but many applications require exact feedback about the actual bit-rate of the compressed quantized output. For example, consider a streaming media system that delivers compressed audio or video information for unbroken playback. An entropy model of the quantized output does not guarantee that actual bit-rate of compressed output satisfies a target bit-rate. If the actual bit-rate of compressed output is much greater than the target bit-rate, playback is disrupted. On the other hand, if the actual bit-rate of compressed output is much lower than the target bit-rate, the quality of the quantized output is not as good as it could be.

The dependency between actual bit-rate of compressed output and quantization threshold is difficult to precisely express -- it depends on complex, non-linear, and dynamic interaction between the entropy of the quantized output and the compression techniques used on the quantized output. The relation changes for different types of data and different compression techniques. Thus, to determine actual bit-rate of compressed, quantized output, the quantized output must be compressed with brute force, computationally expensive and time-consuming operations.

One adaptive quantization technique uses actual bit-rate of compressed output as feedback to find an optimal quantization threshold (highest fidelity to original signal) for a target bit-rate E_{TGT}. For a fixed dynamic range, a binary search quantizer tests candidate quantization thresholds T for a block of input data according to a binary search approach. The process of testing candidate quantization thresholds to find an acceptable quantization threshold is a quantization loop.

The binary search quantizer sets a search range bounded by $T_{HIGH} = T_{MAX}$ and $T_{LOW} = T_{MIN}$. Splitting the search range, the binary search quantizer selects a candidate quantization threshold in the middle $T_{MID} = 0.5(T_{HIGH} + T_{LOW})$ and applies it to the data. The quantized output is compressed. If the resulting actual bit-rate E_{MID} is acceptable, the process stops. Otherwise, the search range is halved and the process repeats. The search range is halved by setting T_{HIGH} to T_{MID} if the actual bit-rate E_{MID} exceeded the target bit-rate E_{TGT}, or by setting T_{LOW} to T_{MID} if the actual bit-rate E_{MID} fell below the target bit-rate E_{TGT}.

In practice, this process also stops if $\left|ceil(\log_L(T_{HIGH})) - ceil(\log_L(T_{LOW}))\right| < 1$, where L is an implementation-dependent constant and ceil(x) is the smallest integer that is greater than or equal to x. This condition reflects a logarithmic dependency between absolute value of T and subjective perception. At higher values of T, humans are less sensitive to changes in T.

Figure 1 is a graph showing the results of a quantization loop with a binary search approach (100). Figure 1 shows a range of quantization thresholds T (110), a range of actual bit-rates E_X (120) of compressed output, and a target bit-rate E_{TGT} (130), which is set at 875 bits. The binary search quantizer starts with quantization thresholds 2 and 34, known to be too small and too large, respectively. The binary search quantizer selects the midpoint quantization threshold 18 and measures the actual bit-rate E_1 of compression operation. As E_1 is far below the target bit-rate E_{TGT}, the quantization threshold 18 becomes the new high bound. The binary search quantizer selects a new midpoint quantization threshold 10, measures the actual bit-rate E_2, and makes the quantization threshold 10 the new high bound. This process continues through the quantization thresholds 6 (resulting actual bit-rate E_3, too high) and 8 (resulting actual bit-rate E_4, too low) before stopping after quantization threshold 7 (resulting actual bit-rate E_5, acceptable).

The binary search approach finds an acceptable quantization threshold within a bounded period of time -- the process stops when the search range becomes small enough. On the other hand, the binary search technique uses 5-8 loop iterations on average, depending on choice of T_{MAX}, T_{MIN}, L and other implementation details in different encoders. Each iteration involves an expensive computation of actual bit-rate of compressed output quantized according to a candidate quantization threshold. In total, these quantization loop iterations take

from 20%-80% of encoding time, depending on the encoder used and bit-rate/quality of the data.

SUMMARY OF THE INVENTION

The present invention reduces the number of iterations of a quantization loop by using a heuristic approach. Reducing the number of iterations instantly improves performance of an encoder system by eliminating computationally-expensive and time-consuming compression operations. Thus, the encoder system can use less expensive hardware, devote resources to other aspects of encoding, reduce delay time in the encoder system, and/or devote resources to other tasks.

To reduce the number of iterations of the quantization loop, a quantizer estimates a quantization threshold for a block of data based upon a heuristic model of actual bit-rate as a function of quantization threshold for a data type. The quantizer evaluates the actual bit-rate of compressed output quantized by the estimated quantization threshold. If the actual bit-rate satisfies a criterion such as proximity to a target bit-rate, the quantizer sets the estimated quantization threshold as the final quantization threshold. Otherwise, the quantizer adjusts the heuristic model and repeats the process with a new estimated quantization threshold.

Additional features and advantages of the invention will be made apparent from the following detailed description of an illustrative embodiment that proceeds with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a graph showing the results of a prior art quantization loop with a binary search approach.

Figure 2 is a block diagram of a computing environment used to implement the illustrative embodiment.

Figure 3 is a block diagram of an encoder system including the quantizer of the illustrative embodiment.

Figure 4 is a flow chart showing a quantization loop with a heuristic approach according to the illustrative embodiment.

Figure 5 is a graph showing the heuristic model of actual bit-rate versus quantization threshold through three iterations of the quantization loop of the illustrative embodiment.

DETAILED DESCRIPTION OF AN ILLUSTRATIVE EMBODIMENT

The illustrative embodiment of the present invention is directed to a quantization loop with a heuristic approach. The heuristic approach reduces iterations of the quantization loop during uniform, scalar quantization of spectral audio data.

The heuristic models actual bit-rate of compressed output as a function of uniform, scalar quantization threshold for a block of data. Initially, the model is parameterized for typical spectral audio data. A quantizer estimates a first

quantization threshold based upon the heuristic model and the spectral energy of a block of spectral audio data.

The quantizer applies the first quantization threshold to the block, which is subsequently compressed by entropy coding. Depending on the actual bit-rate of the compressed output, the quantizer 1) accepts the first quantization threshold or 2) adjusts the heuristic model, estimates a new quantization threshold, and repeats the process. A quantization threshold is acceptable if it results in compressed output with actual bit-rate that falls within a range below a target bit-rate. Other acceptability criterion are possible. For example, an acceptability criterion can be based upon proximity to the target bit-rate, proximity to a target distortion, or distance between quantization thresholds in successive iterations.

The heuristic approach of the present invention can be applied to quantization loops for data other than spectral audio data. For example, after making any appropriate customizations to the heuristic model, a quantizer can process time domain audio data or video data. Although the illustrative embodiment describes uniform, scalar quantization, alternative embodiments apply a quantization loop with a heuristic approach to other quantization techniques.

The quantization loop with a heuristic approach occurs during encoding. During decoding, the compressed output is decompressed in an entropy decoding operation. The decompressed output is dequantized by applying the quantization threshold (earlier used in quantization) to the decompressed output in an inverse quantization operation.

I. Computing Environment

Figure 2 illustrates a generalized example of a suitable computing environment (200) in which the illustrative embodiment may be implemented. The computing environment (200) is not intended to suggest any limitation as to scope of use or functionality of the invention, as the present invention may be implemented in diverse general-purpose or special-purpose computing environments.

With reference to Figure 2, the computing environment (200) includes at least one processing unit (210) and memory (220). In Figure 2, this most basic configuration is included within dashed line (230). The processing unit (210) executes computer-executable instructions and may be a real or a virtual processor. In a multi-processing system, multiple processing units execute computer-executable instructions to increase processing power. The memory (220) may be volatile memory (e.g., registers, cache, RAM), non-volatile memory (e.g., ROM, EEPROM, flash memory, etc.), or some combination of the two. The memory (220) stores software (280) implementing a quantization loop with a heuristic approach for an encoder system.

A computing environment may have additional features. For example, the computing environment (200) includes storage (240), one or more input devices (250), one or more output devices (260), and one or more communication connections (270). An interconnection mechanism (not shown) such as a bus, controller, or network interconnects the components of the computing environment (200). Typically, operating system software (not shown) provides an

operating environment for other software executing in the computing environment (200), and coordinates activities of the components of the computing environment (200).

The storage (240) may be removable or non-removable, and includes magnetic disks, magnetic tapes or cassettes, CD-ROMs, DVDs, or any other medium which can be used to store information and which can be accessed within the computing environment (200). The storage (240) stores instructions for the software (280) implementing the quantization loop with a heuristic approach for an encoder system.

The input device(s) (250) may be a touch input device such as a keyboard, mouse, pen, or trackball, a voice input device, a scanning device, or another device that provides input to the computing environment (200). For audio or video encoding, the input device(s) (250) may be a sound card, video card, or similar device that accepts audio or video input in analog or digital form. The output device(s) (260) may be a display, printer, speaker, or another device that provides output from the computing environment (200).

The communication connection(s) (270) enable communication over a communication medium to another computing entity. The communication medium conveys information such as computer-executable instructions or other data in a modulated data signal. A modulated data signal is a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media include

wired or wireless techniques implemented with an electrical, optical, RF, infrared, acoustic, or other carrier.

The invention can be described in the general context of computer-readable media. Computer-readable media are any available media that can be accessed within a computing environment. By way of example, and not limitation, with the computing environment (200), computer-readable media include memory (220), storage (240), communication media, and combinations of any of the above.

The invention can be described in the general context of computer-executable instructions, such as those included in program modules, being executed in a computing environment on a target real or virtual processor. Generally, program modules include routines, programs, libraries, objects, classes, components, data structures, etc. that perform particular tasks or implement particular abstract data types. The functionality of the program modules may be combined or split between program modules as desired in various embodiments. Computer-executable instructions for program modules may be executed within a local or distributed computing environment.

For the sake of presentation, the detailed description uses terms like "determine," "get," "estimate," and "apply" to describe computer operations in a computing environment. These terms are high-level abstractions for operations performed by a computer, and should not be confused with acts performed by a human being.

II. Encoder System Including Quantizer

Figure 3 is a block diagram of an encoder system (300) including a uniform, scalar quantizer (330). The encoder system receives analog time domain audio data and produces compressed, spectral audio data. The encoder system (300) transmits the compressed output over a network (360) such as the Internet.

An analog to digital converter (310) digitizes analog time domain audio data. Although this digitization is a type of quantization, in the illustrative embodiment the quantization loop occurs later in the encoder system (300).

After or in conjunction with the analog to digital conversion, a time domain to frequency domain transformer (320) converts time domain audio data $A = \{a_1,...,a_n\}$ into frequency domain (i.e., spectral) data $S = \{s_1,...,s_n\}$. Typical transformations include wavelet transforms, Fourier transforms, and subband coding.

The spectral audio data is further processed to emphasize perceptually significant spectral data, a process sometimes called masking. Certain frequency ranges of spectral data (e.g., low frequency ranges) are more significant to a human listener than other frequency ranges (e.g., high frequency ranges). Accordingly, the spectral audio data is processed to make important spectral data more robust to subsequent quantization. Masking uses selective quantization, applying different weights to different ranges of spectral data. The quantization loop can be implemented in conjunction with masking, for example, by modifying a uniform scalar quantization threshold by different weights for different frequency ranges of spectral data according to perceptual significance.

The quantizer (330) quantizes a block of spectral coefficients for audio data held in a buffer (not shown). The quantizer applies a quantization threshold T set through a quantization loop to the block of data, producing quantized output. The quantization loop considers a target bit-rate E_{TGT} (340) that constrains the quantization threshold T. The quantization loop receives feedback (350) indicating the actual bit-rate E_X of compressed output quantized according to a candidate quantization threshold T. Eventually, the quantizer (330) stops after determining a quantization threshold is acceptable. The details of the quantization loop are provided in the following section.

The entropy encoder (360) compresses the quantized output of the quantizer (330). Typical entropy coding techniques include arithmetic coding, Huffman coding, run length coding, LZ coding, and dictionary coding. The actual bit-rate E_X of the compressed block of audio spectral data quantized by the candidate quantization threshold is the basis of feedback (350) in the quantization loop. In Figure 3, the entropy encoder (360) puts compressed output in the buffer (370), and the fullness of the buffer (370) indicates actual bit-rate E_X for feedback (350). The fullness of the buffer (370) can depend on a trait of the input data that affects the efficiency of compression (e.g., uncharacteristically high or low entropy). Alternatively, the fullness of the buffer (370) can depend on the rate at which information is depleted from the buffer (370) for transmission.

Before or after the buffer (370), the compressed output is channel coded for transmission over the network (380). The channel coding can apply error protection and correction data to the compressed output.

A decoder system receives compressed, spectral audio data output by the encoder system (300) and produces analog time domain audio data. In the decoder system, a buffer receives compressed output transmitted over the network (360). An entropy decoder decompresses the compressed output in an entropy decoding operation, producing a block of quantized spectral coefficients for audio data. A dequantizer dequantizes the quantized spectral coefficients in an inverse quantization operation. The inverse quantization operation uses the quantization threshold previously determined to be acceptable by the quantizer (330). A frequency domain to time domain transformer and a digital to analog converter perform the inverse of the operations of the time domain to frequency domain transformer (320) and the analog to digital converter (310), respectively.

III. Quantization Loop with Heuristic Approach

The quantization loop selects candidate quantization thresholds based upon a heuristic model of actual bit-rate versus quantization threshold for a block of data. In the first iteration, the selected quantization threshold often yields compressed output with actual bit-rate acceptably close to the target bit-rate, thereby avoiding subsequent iterations. If not, bit-rate feedback from the first iteration is used to adjust the heuristic model, which improves the second

quantization threshold. Thus, in subsequent iterations, the selected quantization threshold quickly converges on an acceptable quantization threshold.

Figure 4 shows a flowchart (400) for a quantization loop performed by a quantizer. At the start (410), the quantizer gets (420) a block of 1000 spectral coefficients for audio data. Other block sizes and data types are possible. Block size is an implementation decision that balances the goal of optimizing quantization for smaller blocks against the cost of finding a quantization threshold for each block.

The quantizer gets (430) the target bit-rate E_{TGT} for the block of spectral coefficients. The target bit-rate gives the allowable number of bits for the compressed output under current operating constraints. A typical operating constraint is the number of bits that can be streamed over the Internet for unbroken playback, possibly factoring in current levels of network congestion. Another operating constraint could relate to processing capacity of the encoder system or a bit-rate goal for a file including the compressed output.

In the illustrative embodiment, if the actual bit-rate E_X of the final compressed output falls below the target bit-rate E_{TGT}, the unused bit-rate capacity is ignored in quantizing subsequent blocks. Alternatively, extra bits from a previous block are allocated to the target bit-rate for the current block, so long as the average bit-rate over a span of blocks satisfies a bandwidth target to prevent buffer overflow and underflow.

The quantizer sets (440) a heuristic model of actual bit-rate of compressed output versus quantization threshold. The quantizer sets the heuristic model according to a model for spectral audio data, the spectral energy of the block, and any feedback from previous iterations. The quantizer calculates (450) a quantization threshold T based upon the heuristic model and quantizes (460) the block of data using the calculated T. Each spectral coefficient s_i is quantized by T according to the formula:

$$q_i = round\left(\frac{s_i}{2T}\right) \quad ; \qquad (1)$$

where round(x) is the integer nearest to x. Alternatively, another quantization formula is used, for example, one that divides s_i by T instead of $2T$, with corresponding changes to the heuristic model.

The quantizer determines (470) whether the quantization threshold is acceptable. For example, the quantizer compares the actual bit-rate E_X of the compressed output to the target bit-rate E_{TGT} to determine if the actual bit-rate is below but sufficiently close to the target bit-rate. Other acceptability criterion are possible, for example, proximity to the target bit-rate, proximity to a target distortion or distance between quantization thresholds in successive iterations. In an alternative embodiment, the quantizer tests a candidate quantization threshold after finding an acceptable quantization threshold to verify that no better quantization threshold exists. The cost of this extra iteration can be justified if an

application that is extremely sensitive to distortion in the data and the likelihood of finding a better quantization threshold is non-negligible.

If the quantization threshold is acceptable, the quantization loop finishes for that block. If the quantization threshold is not acceptable, the quantization loop again sets (440) the heuristic model, now considering the resulting actual bit-rate from the previous iteration.

After the quantizer finds an acceptable quantization threshold, the quantizer determines (480) whether any more blocks of spectral data remain to be quantized. If so, the quantizer gets (420) the next block and continues from that point. Otherwise, the quantizer finishes (490).

In an alternative embodiment, the quantizer applies different heuristic models to different blocks for blocks that have different statistical characteristics (e.g., blocks of low frequency range spectral data vs. blocks of high frequency range spectral data).

A. Heuristic model for spectral audio data

In the quantization loop, the heuristic model determines an initial quantization threshold and improves selection of subsequent quantization thresholds. The initial parameters of the heuristic model depend on the type of data being compressed, and can be set through training or statistical analysis.

In general, the problem of finding a quantization threshold that is optimal for a target bit-rate cannot be solved a priori due to the complex, non-linear dependencies between the quantized output and the compression techniques used

on the quantized output. For quantization of arbitrary, unknown data, the binary search approach described above may be optimal.

Input signals of a particular data type, however, typically have similarities that can be exploited to tune a quantization loop. For example, one feature of audio (and video) data is that the distribution of spectral data is not uniform. Smaller value spectral data is more frequent that larger value data, and prevails in the output of a quantizer. Table 2 gives a distribution of quantized spectral coefficients for music and speech encoded with a subject audio encoder.

$\|q_i\|$	Frequency of Occurrence	Encoded Size (in bits)
0	78.0%	.75
1	14.5%	2
2	4.5%	4
3	2.0%	6
>3	<1.0%	>6

Table 2: Distribution of quantized spectral data for music and speech

Table 2 gives summary results for several sequences of audio data. For any given block of spectral audio data, the frequencies of occurrence will vary as the quantization threshold varies. For the summary distribution and expected bit-allocation of Table 2, however, the actual bit-rate $E(S,T)$ of a typical block of quantized spectral audio data S is approximately:

$$E(S,T) \approx \sum_{|q_i|=0} 0.75 + \sum_{|q_i|>0} 2|q_i| \ ; \qquad (2)$$

Assuming for the sake of simplicity that spectral coefficients s_i are uniformly distributed in the range $(-T,T)$, corresponding quantized values are:

$$q_i = round\left(\frac{s_i}{2T}\right) \approx \frac{s_i}{2T} \quad ; \qquad (3)$$

q_i is equal to zero if $|s_i| < T$, and the average value of a spectral coefficient quantized to zero is $|s_i| = 0.5T$. Also assuming for the sake of simplicity that the spectral coefficients are uniformly distributed in higher quantization levels as well, by substitution equation (2) becomes:

$$E(S,T) \approx \sum_{|s_i|<T}\left(0.25 + 2\frac{|s_i|}{2T}\right) + \sum_{|s_i|\geq T} 2\frac{|s_i|}{2T} \quad ; \qquad (4)$$

As noted in Table 2, roughly 80% of typical quantized spectral audio data is 0 value. Factoring this observation into equation (4) yields the equation:

$$E(S,T) \approx 0.2N + \frac{1}{T}\sum_{i=0}^{N}|s_i| \quad ; \qquad (5)$$

where N is the number of spectral coefficients in the block. Equation (5) can be expressed more simply as:

$$E(S,T) \approx 0.2N + \frac{|S|}{T} \quad ; \qquad (6)$$

where $|S|$ is the cumulative energy of the spectral coefficients.

While the derivation of equations (2)-(6) depended upon statistical analysis of typical quantized spectral audio data for the subject audio encoder, a generalization of equation (5) can be applied to other forms of data:

$$E(S,T) \approx C_1 N + \frac{C_2}{T}|S| \quad ; \qquad (7)$$

where C_1 and C_2 are implementation-dependent coefficients that can be derived by statistical analysis and $|S|$ is the cumulative energy of the spectral coefficients.

Alternatively, instead of statistical analysis, the coefficients of equations (5) or (7) can be determined through training on a set of typical data. For the subject audio encoder, for example, the coefficients C_1 and C_2 can be determined by minimizing mean square error between actual bit-rates and bit-rates predicted by the heuristic model across a set of representative audio sequences.

B. Iterations of the quantization loop

For an initial approximation T_1 of the final quantization threshold, the quantizer considers the target bit-rate E_{TGT}, the cumulative spectral energy $|S|$ of the block of spectral audio data, and a factor of the number N of spectral coefficients in the block. The quantizer applies this factors to equation (6):

$$T_1 = \frac{|S|}{E_{TGT} - 0.2N} \quad ; \qquad (8)$$

If the actual bit-rate $E(S,T_1)$ of compressed output quantized by the initial approximation T_1 is not acceptable, the quantizer performs one or more additional iterations of the quantization loop.

For a second approximation T_2, the quantizer adjusts the previous approximation T_1 by the proportion by which the first actual bit-rate $E(S,T_1)$

deviated from the target output bit-rate E_{TGT}. The quantizer relates the results of the first iteration to the target bit-rate E_{TGT} and T_2 using the equation:

$$E(S,T) \approx \frac{C}{T}|S| \quad ; \qquad (9)$$

where C is a coefficient relating the first two iterations and $|S|$ is the cumulative energy of the spectral coefficients. Solving equation (9) for C with the results of the first iteration, and then solving equation (9) for T_2 with C and E_{TGT} yields the equation:

$$T_2 = C\frac{|S|}{E_{TGT}} = \frac{T_1 E(S,T_1)}{|S|}\frac{|S|}{E_{TGT}} = \frac{T_1 E(S,T_1)}{E_{TGT}} \quad ; \qquad (10)$$

Alternatively, instead of equations (9) and (10), a modified version of equation (5) can be used to find the second approximation T_2, where the coefficient C modifies the cumulative spectral energy. In experiments, equation (10) gave better results for the second approximation T_2 for spectral audio data than the modified version of equation (5).

If the actual bit-rate $E(S,T_2)$ of compressed output quantized by the second approximation T_2 is not acceptable, the quantizer performs one or more additional iterations of the quantization loop.

For any subsequent iterations, the quantizer approximates a quantization threshold T_k based upon the results of the previous two iterations. The quantizer uses the equation:

$$E(S,T_k) \approx C_1 N + \frac{C_2}{T_k}|S| \ ; \qquad (11)$$

where C_1 and C_2 are deduced from the results of the first two equations. For example, for the third iteration, the results of the first iteration are put in a first equation (11), the results of the second iteration are put in a second equation (11), and the two equations are solved for C_1 and C_2. The values for C_1, C_2, and E_{TGT} are then substituted into equation (11), which is then solved for T_k:

$$T_k = \frac{C_2|S|}{E_{TGT} - C_1 N} \ ; \qquad (12)$$

If the actual bit-rate $E(S,T_k)$ of compressed output quantized by the k-th approximation T_k is not acceptable, the quantizer performs an additional iteration of the quantization loop using equation (12) and coefficients C_1 and C_2 with values deduced from the most recent two iterations.

Figure 5 is a graph (500) showing the heuristic model as it changes through three iterations of the quantization loop. The quantization loop determines a quantization threshold for a block of hypothetical spectral audio data then encoded with a hypothetical audio encoder.

The heuristic model relates actual bit-rate E_X (520) as a function of quantization threshold (510). The target bit-rate E_{TGT} (530) is 875 bits. The quantization loop continues until the actual bit-rate E_X falls within the range (540) of acceptable actual bit-rates under the target bit-rate E_{TGT} (530). In Figure 5, the range (540) includes actual bit-rates up to 3% less than the target bit-rate

(530). So any output bit-rate greater than $875*(1-.03)=849$ bits and less than or equal to 875 bits is acceptable. Other ranges (e.g., 0%, 5%, 7%) are possible. The size of the range is an implementation decision that balances output quality against the costs of the extra iterations needed to achieve the highest possible quality for a target bit-rate.

In Figure 5, the cumulative spectral energy $|S|$ is 3400 for the 1000 coefficients of the input block. The graph for the first iteration (550) shows the following equation based on equation (6), which includes parameters C_1 and C_2 set for typical spectral audio data:

$$E(S,T_1) \approx 200 + \frac{3400}{T_1} \quad ; \qquad (13)$$

Solving equation (13) for T_1 with the target bit-rate E_{TGT} of 875 bits gives a quantization threshold $T_1 = 5.04 \approx 5$. Applying T_1 to the spectral data, however, results in actual bit-rate of 1400 bits for the compressed output.

The graph for the second iteration (560) shows the following equation based on equation (10) and adapted according to the results of the first iteration:

$$E(S,T_2) \approx \frac{5*1400}{T_2} \quad ; \qquad (14)$$

Solving equation (14) for T_2 with the target bit-rate $E_{TGT} = 875$ bits gives a quantization threshold T_2 = 8. Applying T_2 to the spectral data, however, results in actual bit-rate of 700 bits for the compressed output.

The graph for the third iteration (570) shows the following equation based on equation (11) and adapted according to the results of the previous two iterations:

$$E(S,T_3) \approx -0.47*1000 + \frac{2.75*3400}{T_3} \quad ; \qquad (15)$$

Solving this equation for T_3 with the target bit-rate $E_{TGT} = 875$ bits gives a quantization threshold $T_3 = 7$. Applying T_3 to the spectral data results in actual bit-rate of 850 bits, which is within the 3% range (540) of the target bit-rate (530).

In alternative embodiments, a heuristic model with a different number or arrangement of parameters relates actual bit-rate of output following compression to quantization threshold for a block of data.

C. Performance of the quantization loop with heuristic approach

Experiments with the subject audio encoder on a broad selection of speech and music sequences show that equation (8) yields an acceptable quantization threshold in the first iteration 20-40% of the time. In other words, 20-40% of the time, the resultant actual bit-rate $E(S,T_1)$ is close enough below the target output bit-rate E_{TGT} that the quantization loop ceases after the first iteration. When a second iteration is required, equation (10) yields an acceptable quantization threshold in the second iteration about 70% of the time. When a third iteration is

required, equation (12) yields an acceptable quantization threshold in the third iteration about 95% of the time.

Compared to the prior art quantization loop with a binary search approach which requires 5-8 iterations on average (depending on implementation in different encoders), the quantization loop with a heuristic approach requires 2 iterations on average for spectral audio data. The quantization loop with a heuristic approach reduces total encoding time by 5-40%, depending on the encoder used and bit-rate/quality of the data.

Having described and illustrated the principles of my invention with reference to an illustrative embodiment, it will be recognized that the illustrative embodiment can be modified in arrangement and detail without departing from such principles. It should be understood that the programs, processes, or methods described herein are not related or limited to any particular type of computing environment, unless indicated otherwise. Various types of general purpose or specialized computing environments may be used with or perform operations in accordance with the teachings described herein. Elements of the illustrative embodiment shown in software may be implemented in hardware and vice versa. The equations described above represent the results of computer operations in a form that facilitates understanding. The actual computer operations leading to the result of an equation can vary depending on implementation.

In view of the many possible embodiments to which the principles of my invention may be applied, I claim as my invention all such embodiments as may come within the scope and spirit of the following claims and equivalents thereto.